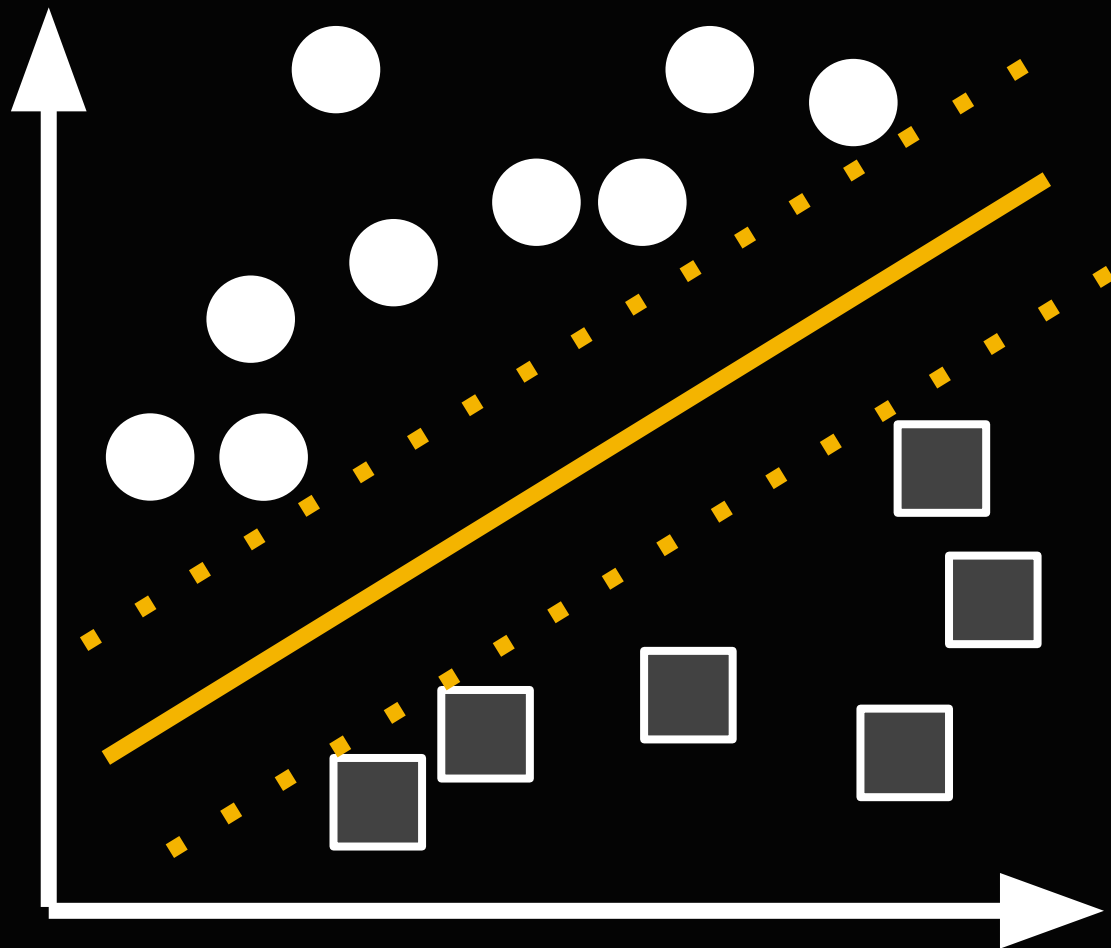
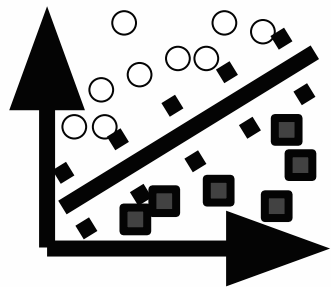
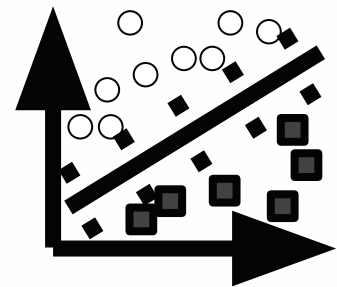


Grid
search
cross
validation



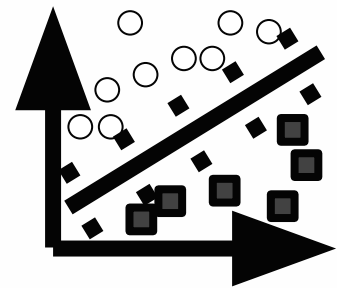
```
grid_search = GridSearchCV(  
    estimator = pipeline,  
    param_grid = param_grid,  
    scoring = "accuracy",  
    cv = 10,  
    n_jobs = -1  
)
```





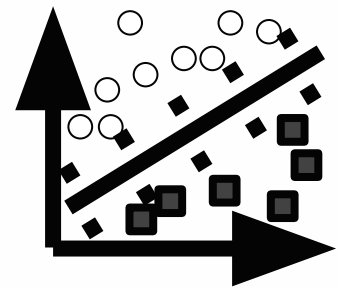
Grid search

- Method to **optimize hyperparameters**
- Find the **best combination** of hyperparameters
for a model and dataset



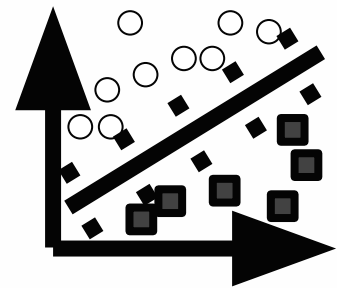
Steps

- Each **combination of parameters** corresponds to 1 model
- Each combination is a **point** on the grid
- **Train** each model and evaluate them with cross validation
- Use the **top performer**



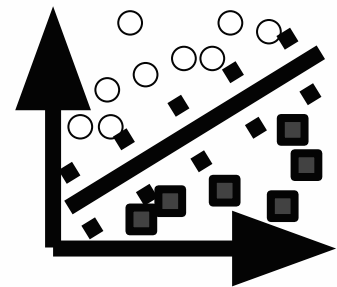
Example: SVM

- Use different combinations of hyperparameters for SVM
 - *Gamma, C, kernel types ...*



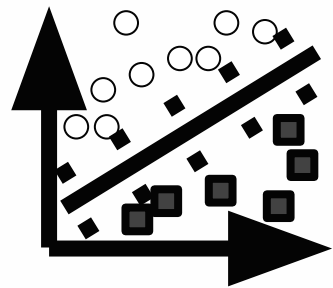
gamma

- size of the radial basis function kernel



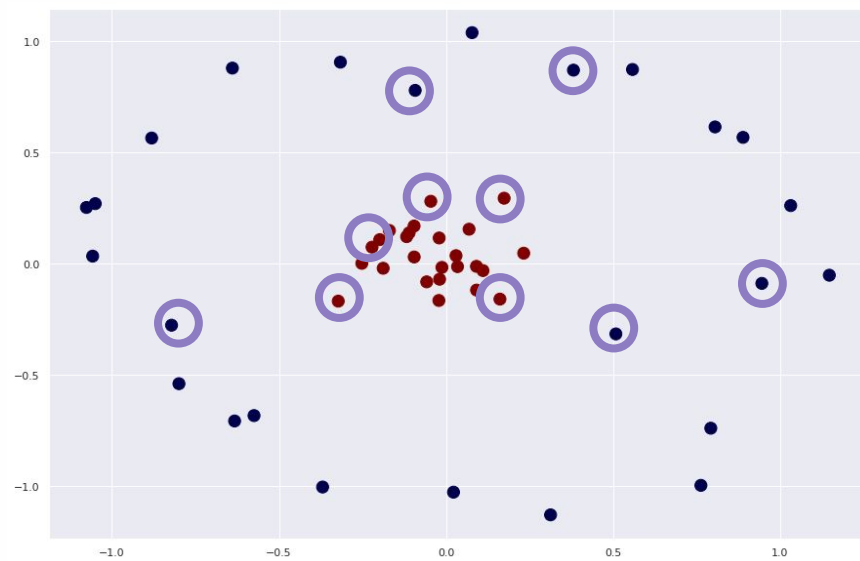
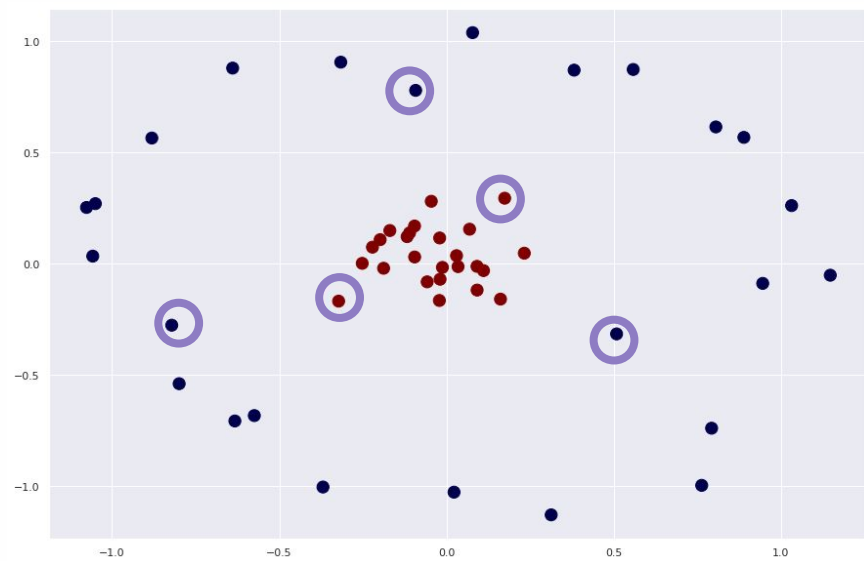
C

- margin hardness
- penalty parameter

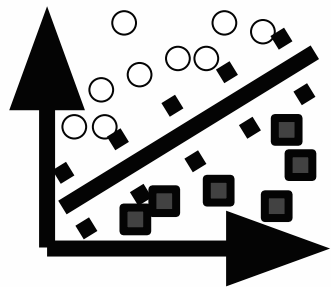


C

- if C is large, datapoints further away from the plane are used in the fit.
 - This can mean overfitting.

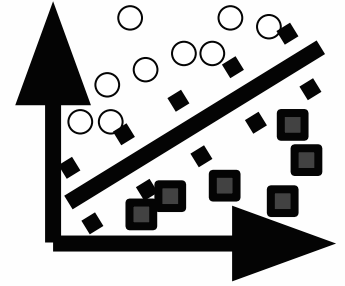


```
param_range = [0.0001, 0.001, 0.01, 0.1, 1.0,  
               10.0, 100.0, 1000.0]
```

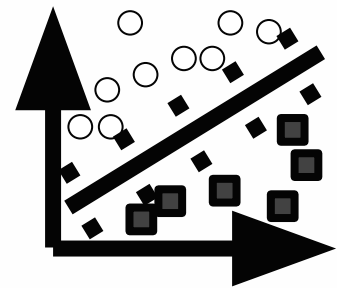


```
param_grid = [{"clf__C": param_range,  
               "clf__kernel": ["linear"]},  
  
              {"clf__C": param_range,  
               "clf__gamma": param_range,  
               "clf__kernel": ["rbf"]}  
]
```

Example: SVM

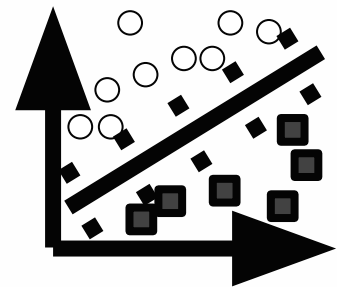


- Grid search **trains a SVM for each parameter** in the range



Pipeline

- assembles steps that can be cross-validated while setting different parameters



```
pipeline = Pipeline(  
    [ ("sc1", StandardScaler()),  
      ("clf", svm.SVC(random_state = 1))  
    ]  
)
```

**Grid search
cross
validation**

